Privacy and Verifiability for Data Storage in Cloud Computing

Melek Önen

August 17, 2015 – IFIP Summer School, Edinburgh
Cloud Computing – Outsourcing storage & computation

- High availability
- No IT maintenance cost
- Decreased Costs
- Elasticity & Flexibility
CSA’s Notorious Nine

Data breaches
- Unauthorized access to client data

Data Loss
- Accidental or malicious destruction

Account hijacking
- Stolen credentials

Insecure APIs
- Adversary steals data from cloud

Denial of Service
- Intolerable system slowdown

Malicious insiders
- More powerful attackers

Abuse of cloud services
- Adversary rents the cloud

Insufficient due diligence
- Mismatched expectations

Shared technology issues
- Adversary breaks out of the hypervisor
Clouds as Adversaries

To trust or how to trust?

- Security Models & Requirements
  - Honest but curious
    - Confidentiality & Privacy
  - Malicious
    - Privacy + Integrity & Transparency

- Challenge

Privacy & Integrity

Big Data

Do not cancel cloud advantages

Lightweight operations at client side
Privacy and verifiability for cloud computing

- Privacy and verifiability for data outsourcing
  - Data encryption with storage efficiency
  - Privacy preserving de-duplication
  - Efficient verification of data storage
  - Proofs of Retrievability

- Privacy and verifiability for data processing
  - Computation over encrypted data
    - Privacy preserving word search
  - Verifiable computation
    - Verifiable word search
Storage efficiency with deduplication

- Storing duplicate data only once
- Total space savings up to 90-95% in backup applications
Conflict with privacy

D = Hello World

ENCRYPTION with K1

owhfg0wgr[w hfrw0[h0[ergh e0[gh0[eg

ENCRYPTION with K2

dfj;dbfrwbfirbf roepthwobgfr ugtwertgrtwu
One solution: Convergent encryption

- Key = hash(data)

- Vulnerability to dictionary attacks
Secure deduplication based on popularity

- Different protection based on popularity
  - Popular data $\rightarrow$ belongs to $\geq t$ users
    - less secure protection
    - to be deduplicated
    - use convergent encryption
  - Unpopular data $\rightarrow$ likely to be unique, highly sensitive
    - stronger protection
    - no need for deduplication
    - use stronger encryption

- Related work [Janek et. al.’ 2014]
  - Combination of convergent encryption with threshold encryption
    - Significant storage and bandwidth overhead
Idea: Secure popularity detection

- **Data protection solutions**
  - Popular data → convergent encryption (CE)
  - Unpopular data → symmetric encryption (SE)

- **Idea: Check popularity**
  - Encrypt “B” with CE
  - lookup for CE(B)
    - CE(B) exists ⇒ deduplication
    - CE(B) does not exist ⇒ encrypt “B” with SE

- **Requirement: secure lookup**
  - if B unpopular, popularity check should not reveal B
  ⇒ Secure lookup based on perfect hashing
Privacy and verifiability for cloud computing

- Privacy and verifiability for data outsourcing
  - Data encryption with storage efficiency
    - Privacy preserving de-duplication
  - Efficient verification of data storage
    - Proofs of Retrievability

- Privacy and verifiability for data processing
  - Computation over encrypted data
    - Privacy preserving word search
  - Verifiable computation
    - Verifiable word search
Verification of data storage

- Motivating scenario: outsourced storage

- Requirements for evidence of correct storage
  - Unlimited integrity checks by client
  - No data transfer
  - Efficient verification
Proofs of Retrievability: Related Work

- **Deterministic solutions**  
  - Costly verification performed on the entire data  
  
- **Probabilistic solutions**  
  - Homomorphic tags  
    - Optimized communication costs  
    - Expensive preprocessing of the file before outsourcing  
    - Significant storage overhead  
  - Randomly located sentinels  
    - Light computation at the client, small storage overhead  
    - Limited number of checks  

[Deswarte et.al, Filho et.al, ..]  
[Ateniese et.al., Juels et.al., Shacham et.al...]
Proofs of Retrievability: StealthGuard

[ESORICS’14]
Proofs of Retrievability: StealthGuard

Word Search Query (Privacy Preserving)

Missing word

Missing data split
Privacy and verifiability for cloud computing

- Privacy and verifiability for data outsourcing
  - Data encryption with storage efficiency
    - Privacy preserving de-duplication
  - Efficient verification of data storage
    - Proofs of Retrievability

- Privacy and verifiability for data processing
  - Computation over encrypted data
    - Privacy preserving word search
  - Verifiable computation
    - Verifiable word search
Privacy preserving word search

Outsourced DB Service

- several years’ corporate data
- regularly stored in the Cloud
- Privacy $\Rightarrow$ Encryption by the customer
- Query: search for word
- How to find it without downloading the entire DB?

Requirement for a new solution

- to search words in an encrypted DB
- with privacy
Privacy requirements

- **Storage privacy**
  - No information on stored data
  - → need for data encryption

- **Query privacy**
  - No information on the targeted word
  - → query privacy, query indistinguishability
  - No information on the result
  - → response privacy

- **Related work**
  - Mostly focused on the query privacy aspect only
  - Query expressiveness
Our approach

- **Querier = Owner**
  - Data privacy AND Query privacy

- **Querier = authorized third party**
  - Data privacy AND Query privacy
  - On-the-fly authorization & revocation

- **Multi-Querier, Multi-Owner**
  - Data privacy AND Query Privacy (becomes mandatory!)
  - On-the-fly authorization/revocation
  - scalability
PRIVACY PRESERVING WORD SEARCH IN MAPREDUCE

- Semantically secure encryption
  - Data privacy

- Word search transformed to PIR problems (single bit)
  - Query privacy

- Parallelism with MapReduce
  - Map: Evaluate small PIR problem on each InputSplit
  - Reduce: combine mapper output with simple addition
PRISM - Upload

Privacy and Verifiability for data storage in cloud computing
Melek Önen
IFIP Summer School, August 2015

File

\[ w_0, w_1, \ldots, w_{n-1} \]

WordEncrypt

\[ E(w_0), E(w_1), \ldots, E(w_{n-1}) \]

Upload

User

Cloud

Mapper

Binary Map

\[ E(w_i), \ldots, E(w_j) \]

\[ E(w_k), \ldots, E(w_l) \]

\[ E(w_m), \ldots, E(w_n) \]

\[ E(w_u), \ldots, E(w_v) \]

\[ \text{hash}(E(w_i)) \]

\[ X, Y, b \]

\[ X \]

\[ Y \]

\[ b \]
**PRISM – Word Search**

Query for word $w$

$$\text{hash}(E(w)) \rightarrow X,Y, b \downarrow$$

PIR query for $(X,Y)$

$$\text{Binary Map} \times$$

$$\text{PIR}(X,Y) \times$$

$$\text{PIR}(X,Y) \times$$

$$\text{PIR}(X,Y) \times$$

$$\text{homomorphic } \sum$$

$$E(\text{result})$$

Mapper

Cloud

Reducer

User
Application Scenario: Auditing with privacy

- Transparency ⇒ Logging
- Logs can be outsourced to the cloud
- User privacy ⇒ Encrypted logs
- Accountability ⇒ Auditor searches on encrypted logs
- Revoke authorized querier at any point of time

Idea

- Privacy preserving word search
  - Combine PIR with Cuckoo Hashing
- Third party authorization and revocation
  - Combine ABE with OPRF
MUSE: Multi-User Searchable Encryption

- User A
- User B
- User Z

Cloud

- Recommender systems
- Smart metering

- Huge number of users (writers) AND queriers (Readers)
- No coordination/trust among readers and writers
Idea: Proxy-based privacy preserving lookup

[ISC’15]

- Privacy against cloud
  - Based on PIR
- Privacy against Proxy
  - Based on underlying bilinear encryption
Privacy and verifiability for cloud computing

- Privacy and verifiability for data outsourcing
  - Data encryption with storage efficiency
    - Privacy preserving de-duplication
  - Efficient verification of data storage
    - Proofs of Retrievability

- Privacy and verifiability for data processing
  - Computation over encrypted data
    - Privacy preserving word search
  - Verifiable computation
    - Verifiable word search
Verifiable word search

- Motivating scenario: outsourced anonymized data

- Requirements for proof of search response
  - Public delegatability: anyone can issue a query
  - Public verifiability: anyone can verify the response
  - Efficient verification
Verifiable test of membership

- Define $P$ for each file $f$ such that:
  $$P(w) = 0 \text{ if } w \in f$$
- Compute public accumulator
  $$\text{Acc}(f) = g^{P(\alpha)}$$
- Proof of membership
  result $P(w), \text{witness } \Omega_w = g^{Q_w(\alpha)}$ s.t.
  $$P(\alpha) = (\alpha - w).Q_w(\alpha) + P(w)$$
- Verification
  $$e(\Omega_w, g^{\alpha}g^{-w}) e(g^{P(w)}, g) = e(\text{Acc}(f), g)$$

$\Rightarrow$ cost increases with # words, one polynomial per file

$\Rightarrow$ use Merkle trees
Conclusion

- **Privacy preserving storage & computation**
  - Privacy preserving word search
  - Privacy preserving deduplication

- **Verifiable storage & computation**
  - Verifiable word search
  - Proof of retrievability
  - Data integrity

- **Next?**
  - Efficient FHE solutions
  - Meanwhile, continue research on dedicated primitives
  - Unified framework

Do not cancel cloud advantages

Big Data

Lightweight operations at client side

Melek Önen
IFIP Summer School, August 2015
Acknowledgments

- **EU Projects**

- **Co-Authors**
  - Monir Azraoui, Pasquale Puzio, Cédric Van Rompay
  - Kaoutar Elkhiyaoui, Erik Blass, Roberto di Pietro, Refik Molva
**Papers**

- *Publicly verifiable conjunctive keyword search in outsourcing database*, M. Azraoui, K. Elkhiyaoui, M. Önen, R. Molva, SPC 2015,
- *Multi-user searchable encryption in the cloud*, C. Van Rompay, R. Molva, M. Önen, ISC’15
- *Privacy preserving delegated word search*, K. Elkhiyaoui, M. Önen, R. Molva, SECRIPT 2014
THANK YOU

melek.onen@eurecom.fr